

# Программирование Python

## Лекция 9

Кафедра ИВТ и ПМ

2017



# План

## Прошлые темы

## Математические пакеты

- Тип array

- Линейная алгебра

- Интерполяция

- Минимизация



# Outline

## Прошлые темы

### Математические пакеты

Тип array

Линейная алгебра

Интерполяция

Минимизация



# Прошлые темы

- ▶ Что такое пакет?
- ▶ Для чего нужна программа `pip`?
- ▶ Как её запустить?



# Прошлые темы

- ▶ Что такое пакет?
- ▶ Для чего нужна программа pip?
- ▶ Как её запустить?
  - ▶ из терминала Anaconda Prompt
  - ▶ из командной строки Windows, из каталога, где установлен Python.
- ▶ Как посмотреть список установленных python пакетов?



# Прошлые темы

- ▶ Что такое пакет?
- ▶ Для чего нужна программа pip?
- ▶ Как её запустить?
  - ▶ из терминала Anaconda Prompt
  - ▶ из командной строки Windows, из каталога, где установлен Python.
- ▶ Как посмотреть список установленных python пакетов?  
`pip list`



# Прошлые темы

- ▶ Что такое репозиторий Python?



## Прошлые темы

- ▶ Что такое репозиторий Python?  
Хранилище Python пакетов.
- ▶ Как получить доступ к этому репозитарию?



## Прошлые темы

- ▶ Что такое репозиторий Python?  
Хранилище Python пакетов.
- ▶ Как получить доступ к этому репозитарию?  
С помощью программы pip.
- ▶ Как установить пакет?

# Прошлые темы

- ▶ Что такое репозиторий Python?  
Хранилище Python пакетов.
- ▶ Как получить доступ к этому репозитарию?  
С помощью программы `pip`.
- ▶ Как установить пакет?  
`pip install имяпакета`
- ▶ Как удалить пакет?



## Прошлые темы

- ▶ Что такое репозиторий Python?  
Хранилище Python пакетов.
- ▶ Как получить доступ к этому репозитарию?  
С помощью программы `pip`.

- ▶ Как установить пакет?  
`pip install имяпакета`

- ▶ Как удалить пакет?  
`pip uninstall имяпакета`



# Прошлые темы

► `pip install numpu`

Collecting numpu

Could not find a version that satisfies the requirement numpu  
(from versions: )

No matching distribution found for numpu

О какой проблеме сообщает pip?



# Прошлые темы

► `pip install numpu`

Collecting numpu

Could not find a version that satisfies the requirement numpu  
(from versions: )

No matching distribution found for numpu

О какой проблеме сообщает `pip`?

Пакета с именем *numpu* в репозитории python не существует.



# Outline

Прошлые темы

## Математические пакеты

Тип array

Линейная алгебра

Интерполяция

Минимизация



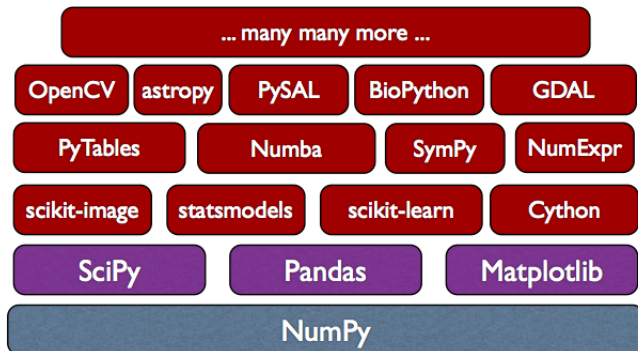
# Популярные математические пакеты

- ▶ **numpy** - работа с матрицами и многомерными массивами;
- ▶ **scipy** - научные и инженерные вычисления, использует numpy;
- ▶ **matplotlib** - построение графиков и диаграмм;
- ▶ **seaborn** - визуализация статистических данных, эстетичнее чем matplotlib;
- ▶ **mpld3** - создание интерактивных графиков (использует D3.js, графики показываются в окне браузера);
- ▶ **pandas** - анализ данных: статистики, регрессия, визуализация и т.п.



# Популярные математические пакеты

Многие пакеты построены на основе NumPy





# Numpy и SciPy

**NumPy** (NUMerical PYthon) - пакет содержащий типы и математические алгоритмы, оптимизированные для работы с многомерными наборами данных.

**SciPy** (Science Python) — библиотека для языка программирования Python с открытым исходным кодом, предназначенная для выполнения научных и инженерных расчётов (включает в себя NumPy).

SciPy можно рассматривать как альтернативу MATLAB <sup>1</sup>.

---

<sup>1</sup>пакет прикладных программ для решения задач технических вычислений и одноимённый язык программирования, используемый в этом пакете



## Возможности

- ▶ поиск минимумов и максимумов функций;
- ▶ вычисление интегралов функций;
- ▶ поддержка специальных функций;
- ▶ обработка сигналов;
- ▶ обработка изображений;
- ▶ работа с генетическими алгоритмами;
- ▶ решение обыкновенных дифференциальных уравнений;
- ▶ и др.

Для визуализации результатов расчётов часто применяется библиотека Matplotlib



## Некоторые модули Scipy

- ▶ **constants.** Физические константы и коэффициенты пересчёта (с версии 0.7.0[2]).
- ▶ **integrate.** Инструменты для интегрирования.
- ▶ **interpolate.** Инструменты для интерполяции.
- ▶ **linalg.** Линейная алгебра.
- ▶ **optimize.** Средства оптимизации <sup>2</sup>.
- ▶ **stats.** Статистические функции.

---

<sup>2</sup>нахождения экстремума (минимума или максимума) целевой функции



# Outline

Прошлые темы

Математические пакеты

Тип array

Линейная алгебра

Интерполяция

Минимизация



# Тип данных array

Для работы с векторами, матрицами (в т.ч. многомерными) в numpy используется тип данных **array**.

Размерность array определяется на этапе задания переменной этого типа.

Array можно создать на основе списка любой размерности.

```
# В следующих примерах будем предполагать,  
# что пакет numpy уже подключен  
from numpy import *
```

```
vector = array( [34, -2, 88] ) # создание вектора
```

array похож на список, но с дополнительными методами и операциями умножения, сложения и т.д.



## Тип данных array

Другие способы создания вектора **array**.

- ▶ Набор значений с заданным шагом

`linspace(from, to, step)`

*from* - начальное значение

*to* - конечное значение (не включается в вектор)

*step* - шаг (может быть дробным), по-умолчанию - 1

- ▶ Набор из N значений

`linspace(from, to, N)`

*N* - число значений, на которые будет разделен отрезок  
*[from, to]*

*From* будет включен в результат.



## Тип данных array

- ▶ Набор значений с заданным шагом

```
arange(0, pi, 0.5)
```

```
array([ 0. , 0.5, 1. , 1.5, 2. , 2.5, 3. ])
```

- ▶ linspace(0,10,10)

```
array([ 0. , 1.11111111, 2.22222222, 3.33333333, 4.44444444,  
5.55555556, 6.66666667, 7.77777778, 8.88888889, 10. ])
```



# Outline

Прошлые темы

## Математические пакеты

Тип array

Линейная алгебра

Интерполяция

Минимизация





# Тип данных array

## Создание матрицы

- ▶ Создание матрицы из двумерного списка

```
A = array([[1,2,3],  
          [4,5,6],  
          [7,8,9]])
```



# Тип данных array

## Создание матрицы

- ▶ Создание матрицы заполненной нулями.

```
A = zeroes( размерность )
```

*размерность* - список элементы которого задают размерность матрицы.

```
A = zeros([3, 2])
```

```
array([[ 0.,  0.],  
       [ 0.,  0.],  
       [ 0.,  0.]])
```

- ▶ Создание матрицы заполненной единицами.

```
A = ones( размерность )
```

- ▶ Создание матрицы заполненной случайными числами.

```
A = random( размерность )
```



## Тип данных array

- Создание единичной матрицы.

```
A = eye( размер )
```

*размер* задаётся одним числом, т.к. матрица квадратная.

```
eye( 3 )
```

```
array([[ 1.,  0.,  0.],  
       [ 0.,  1.,  0.],  
       [ 0.,  0.,  1.]])
```



## Тип данных array

- Сохранение в текстовый файл

```
a = np.array([[ 1.,  0.,  0.],  
              [ 0.,  1.,  0.],  
              [ 0.,  0.,  1.]])
```

```
np.savetxt("foo.csv", a, delimiter=",", fmt = "%10.3f")
```

*# 'foo.csv' - имя файла*

*# sep = ',' - символ отделяющий одно число от другого*

*# формат: 10 - число позиций (символов на число)*

*# 3 - число десятичных знаков*

*# f - представление в виде вещественного числа*

*# общее число символов на число можно не указывать*

```
np.savetxt("foo.csv", a, delimiter=",", fmt = "%.3f")
```

stackoverflow: dump a numpy array into a csv file



# Тип данных array

- Загрузка из текстового файла

```
a = np.genfromtxt("foo.csv", delimiter = ",")
```

*# foo.csv - имя файла*

*# delimiter = "," - символ отделяющий одно число от другого*

[stackoverflow.com](https://stackoverflow.com) how to read csv into record array in numpy



# Линейная алгебра

Операции с типом array.

*# умножение на число*

A \* 3.14

```
> array([[ 3.14,  6.28,  9.42],  
        [ 9.42,  6.28,  3.14],  
        [ 6.28,  3.14,  9.42]])
```

*# Умножение на вектор*

A \* [2,3,4]

```
array([[ 2,  6, 12],  
       [ 6,  6,  4],  
       [ 4,  3, 12]])
```

Причём вектор не обязательно должен быть представлен типом array.



# Линейная алгебра

# Сложение матриц

$$A + B$$

# Умножение матриц

$$A @ B$$

# или

$$\text{dot}(A, B)$$



# Линейная алгебра

*# Вычисление определителя*

```
linalg.det(A)
```

*# Вычисление обратной матрицы*

```
linalg.inv(A)
```





# Линейная алгебра

## Решение СЛАУ

СЛАУ - система линейных алгебраических уравнений

Представим СЛАУ в виде матричного уравнения:

$$AX = B$$

где  $A$  - матрица коэффициентов,

$B$  - столбец свободных членов,

$X$  - искомые.



# Линейная алгебра

## Решение СЛАУ

```
# матрица коэффициентов
```

```
A = np.matrix([  
    [1, 2, 3],  
    [3, 2, 1],  
    [2, 3, 1]])
```

```
# столбец свободных членов
```

```
B = np.matrix([[1], [2], [3]])
```

Стоит обратить внимание на то, что вектор-столбец определяется как матрица из одного столбца (список из списков), а не как список из чисел.



# Линейная алгебра

## Решение СЛАУ

```
X = np.linalg.solve(A, B)
```

```
# Настройка вывода. Число знаков после запятой - 4.
```

```
# не выводить числа в экспоненциальной форме
```

```
set_printoptions(precision=4, suppress=True)
```

```
print( X )
```

```
matrix([[ 0.0833],  
        [ 1.0833],  
        [-0.4167]])
```



# Outline

Прошлые темы

## Математические пакеты

Тип array

Линейная алгебра

**Интерполяция**

Минимизация



# Интерполяция

**Интерполяция** - нахождение промежуточных значений величины по имеющемуся дискретному набору известных значений.

Пусть функция задана в табличном виде - набором значения  $X$  и  $Y$ .

Задача - определить значение функции для  $X$ , не перечисленного в таблице.

Применем для этого интерполяцию, построив функцию в аналитическом виде, которая будет проходить наиболее близко к заданным точкам.



# Интерполяция

```
from scipy.interpolate import interp1d
```

```
X = [1,2,3,4,5,6,7]
```

```
Y = [1,4,9,16,25,36,49]
```

```
func = interp1d(X, Y, kind='cubic')
```

func - функция определённая на отрезке от  $\min(x)$  до  $\max(x)$ .

kind - тип интерполяции (см. справку)

В примере использована интерполяция сплайном третьего порядка.

Теперь можно вычислять значение функции в любой точке.

```
func(5.3)
```

```
>>> array(28.090000000000003)
```



# Outline

Прошлые темы

## Математические пакеты

Тип array

Линейная алгебра

Интерполяция

**Минимизация**



# Минимизация функции

Поиск минимума функции.

```
from scipy.optimize import minimize

# определим функцию одной переменной вида  $y = x^3 - 5x$ 
def my_func( x ):
    return (x-3)*(x-3) - 5

x0 = 100 # начальное значение

# для поиска минимума передадим функции minimize
# в качестве параметров исследуемую функцию
# и точку, откуда будет произведён поиск минимума.
minimize(my_func, x0)
```





# Минимизация функции

Вывод:

```
nfev: 12
jac: array([ 0.])
message: 'Optimization terminated successfully.'
fun: -5.0
success: True
x: array([ 3.])
njev: 4
hess_inv: array([[ 0.5]])
status: 0
```

Минимум найден:

$$f(3) = -5$$



# Ссылки и литература

Ссылка на слайды

[github.com/VetrovSV/Programming](https://github.com/VetrovSV/Programming)

